

DETC98/CIE-6006

## HIGH-LIFT OPTIMIZATION DESIGN USING NEURAL NETWORKS ON A MULTI-ELEMENT AIRFOIL

**Roxana M. Greenman**  
 Aerospace Engineer

NASA Ames Research Center  
 Moffett Field, California 94035-1000, U. S. A.  
 Tel: 650-604-3997, Fax: 650-604-2238  
 E-mail: rgreenman@mail.arc.nasa.gov

**Karlin R. Roth**  
 Aerospace Engineer

NASA Ames Research Center  
 Moffett Field, California 94035-1000, U. S. A.  
 Tel: 650-604-6678, Fax: 650-604-2238  
 E-mail: kroth@mail.arc.nasa.gov

### ABSTRACT

The high-lift performance of a multi-element airfoil was optimized by using neural-net predictions that were trained using a computational data set. The numerical data was generated using a two-dimensional, incompressible, Navier-Stokes algorithm with the Spalart-Allmaras turbulence model. Because it is difficult to predict maximum lift for high-lift systems, an empirically-based maximum lift criteria was used in this study to determine both the maximum lift and the angle at which it occurs. Multiple input, single output networks were trained using the NASA Ames variation of the Levenberg-Marquardt algorithm for each of the aerodynamic coefficients (lift, drag, and moment). The artificial neural networks were integrated with a gradient-based optimizer. Using independent numerical simulations and experimental data for this high-lift configuration, it was shown that this design process successfully optimized flap deflection, gap, overlap, and angle of attack to maximize lift. Once the neural networks were trained and integrated with the optimizer, minimal additional computer resources were required to perform optimization runs with different initial conditions and parameters. Applying the neural networks within the high-lift rigging optimization process reduced the amount of computational time and resources by 83% compared with traditional gradient-based optimization procedures for multiple optimization runs.

### NOMENCLATURE

$C_d$  drag coefficient,  $C_d \equiv D/(q_\infty c)$   
 $C_l$  lift coefficient,  $C_l \equiv L/(q_\infty c)$   
 $C_m$  moment coefficient,  $C_m \equiv M/(q_\infty c^2)$

$C_p$  pressure coefficient,  $C_p \equiv (P - P_\infty)/q_\infty$   
 $C_{p\,diff}$  pressure difference  
 $c$  chord  
 $D$  drag force  
 $L$  lift force  
 $M$  pitching moment  
 $ol$  overlap  
 $q_\infty$  freestream dynamic pressure,  $q_\infty \equiv \frac{1}{2}\rho_\infty V_\infty^2$   
 $rms$  root-mean-square  
 $Re_c$  Reynolds number,  $Re_c = \frac{\rho_\infty V_\infty c}{\mu_\infty}$   
 $V$  freestream velocity  
 $\alpha$  angle of attack  
 $\delta$  deflection angle  
 $\mu$  coefficient of viscosity  
 $\rho$  density

### Subscripts

$f$  flap  
 $max$  maximum  
 $s$  slat  
 $\infty$  freestream value

### INTRODUCTION

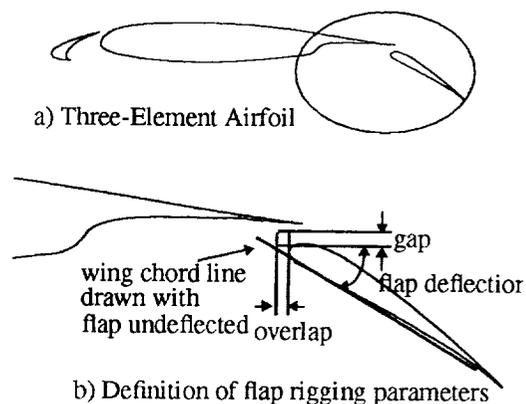
The design of an aircraft's high-lift system is a crucial part of the design phase of commercial and military airplanes since this system controls the takeoff and landing performance. The importance of a well designed high-lift system is seen by increased payloads which also increase the operational

flexibility by extending ranges and by decreasing take-off and landing distances. Traditionally, high-lift designs have been accomplished by extensive wind tunnel and flight test programs which are expensive and difficult due to the large design space. Recently, computational fluid dynamics (CFD) has been incorporated in high-lift design (Ying, 1996). For high-lift applications, CFD can also be expensive because the entire design space is large, grids must be generated around geometrically-complex high-lift devices, and complex flow phenomena must be resolved. In order to achieve optimum, rapid designs, new tools for speedy and efficient analysis of high-lift configurations are required. For these tools to be effective, they need to be functional in all areas of design including wind tunnel, CFD, and flight.

Artificial neural networks are a collection (or network) of simple computational devices which are modeled after the architecture of biological nervous systems. The ability of neural networks to accurately learn and predict nonlinear multiple input and output relationships makes them a promising technique in modeling nonlinear aerodynamic data. Computational fluid dynamics in conjunction with neural networks and optimization may help reduce the time and resources needed to accurately define the optimal aerodynamics of an aircraft including high-lift. Essentially, the neural networks will reduce the amount of data required to define the aerodynamic characteristics of an aircraft while the optimizer will allow the design space to be easily searched for extrema.

Recently, neural networks have been applied to a wide range of problems in the aerospace industry. For example, neural networks have been used in aerodynamic performance optimization of rotor blade design (LaMarsh et al., 1992). The study demonstrated that for several rotor blade designs, neural networks were advantageous in reducing the time required for the optimization. Faller and Schreck (1995) successfully used neural networks to predict real-time three-dimensional unsteady separated flowfields and aerodynamic coefficients of a pitching wing. It has also been demonstrated that neural networks are capable of predicting measured data with sufficient accuracy to enable identification of instrumentation system degradation (McMillen et al., 1995). Steck and Rokhsaz (1997) demonstrated that neural networks can be successfully trained to predict aerodynamic forces with sufficient accuracy for design and modeling. Rai and Madavan (1998) demonstrated the feasibility of applying neural networks to aerodynamic design of turbomachinery airfoils.

Neural networks have been used at NASA Ames Research Center to minimize the amount of data required to define the aerodynamic performance characteristics of a wind tunnel model (Jorgensen and Ross, 1997 and Ross et al., 1997). It was shown that when only 50% of the data acquired from the wind



**Figure 1** Flap-Edge Geometry.

tunnel test was used to train neural nets, the results had a predictive accuracy equal to or better than the experimental data. The success of the NASA Ames neural network application for wind-tunnel data prompted this current study (Greenman, 1998) to use optimization with neural networks to optimize high-lift aerodynamics of a multi-element airfoil.

This paper describes a process which allows CFD to impact high-lift design. This process has three phases: 1) generation of the training database using CFD; 2) training of the neural networks; and 3) integration of the trained neural networks with an optimizer to capture and search the high-lift design space. In this study, an incompressible two-dimensional Navier-Stokes solver is used to compute the flowfield about the three-element airfoil shown in Figure 1. The selected airfoil is a cross-section of the Flap-Edge model (Storms, 1997) that was tested in the 7-by 10-Foot Wind Tunnel No. 1 at the NASA Ames Research Center. Within the CFD database for this flap optimization problem, there are two different slat deflection settings and for each of these, 27 different flap riggings (refer to Figure 1b) are computed for ten different angles of attack. The neural networks are trained by using the flap riggings and angles of attack as the inputs and the aerodynamic forces as the outputs. The neural networks are defined to be successfully trained to predict the aerodynamic coefficients when given a set of inputs that are not in the training set, the outputs are predicted within the experimental error. The experimental error of the total lift coefficient ( $C_l$ ) is  $\pm 0.02$  for  $C_l \leq 0.95C_{l_{max}}$  and  $\pm 0.06$  for  $C_l \geq 0.95C_{l_{max}}$ . Finally, the trained neural networks are integrated with the optimizer to allow the design space to be easily searched for points of interest. It will be shown that this enhanced design process minimizes the cost and time required to accurately optimize the high-lift flap rigging.

A brief description of the training set generation is presented in the next section, including grid generation, the governing equations, maximum lift criteria, and the flow solver. Next, the neural network training is discussed followed by the optimi-

zation process. The results are then presented, from which the effectiveness of optimization with neural networks as a tool to reduce resources required in aerodynamic design is discussed.

## TRAINING SET GENERATION

### Geometry Definition

Extensive wind-tunnel investigations (Storms, 1997) have been carried out for the Flap-Edge geometry shown in Figure 1. The model is a three-element airfoil consisting of a 12%*c* LB-546 slat, NACA 63<sub>2</sub>-215 Mod B main element and a 30%*c* Fowler flap where *c* is chord and is equal to 0.761 m (30.0 inches) for the undeflected (clean, all high-lift components stowed) airfoil. As mentioned, two-different slat deflection angles that are computed, six and twenty-six degrees. Each slat has a gap of  $gap_s = 2.0\%c$  and an overlap of  $ol_s = -0.05\%c$ . In this present study, only the results of the six-degree slat deflection data set are presented (detailed results for  $\delta_s = 26.0^\circ$  are presented by Greenman (1998)). For the computational data base, 27 different flap riggings are created for each slat configuration. The flap riggings are combinations of the following flap deflection, gap, and overlap defined in Figure 1b. The flap deflection angles are  $\delta_f = 25.0^\circ, 29.0^\circ, \text{ and } 39.5^\circ$ . The three gap settings are  $gap_f = 1.50, 2.10, \text{ and } 2.70\%c$  whereas the overlap settings are  $ol_f = 0.40, 1.00, \text{ and } 1.50\%c$ . All gap and overlap values in this paper are expressed in terms of percent clean chord, %*c*. The range of angle of attack varies from  $0.0^\circ \leq \alpha \leq 22.0^\circ$  and  $Re_c = 3.7$  million in this study.

### Grid Generation

The grids around the three-element airfoil are generated using OVERMAGG (Rogers, 1997) which is an automated script system used to perform overset multi-element airfoil grid generation. OVERMAGG takes as input the surface definition of the individual elements of the airfoil. Then it creates a surface grid for each individual element by generating and redistributing points from the given surface definition. It calls the HYPGEN code (Chan et al., 1993) to generate volume grids about each element. The finite difference volume grid is generated in the normal direction of the surface by solving a set of hyperbolic partial differential equations. OVERMAGG also automatically calls the PEGSUS code (Suhs and Tramel, 1991) to unite the individual meshes into an overset grid system which is the final output of OVERMAGG.

Figure 2 shows the grid system that is used. A grid resolution study (Greenman, 1998) is conducted to determine the grid density required to solve the physical flow features. As a result, a total of 121,154 grid points are used consisting of a 242 x 81 C-grid around the slat; a 451 x 131 C-grid around the main element; and a 351 x 121 embedded grid around the flap which is used to help resolve the merging wake in this region. The normal wall spacing for all grids is  $5 \times 10^{-6}$  chords.

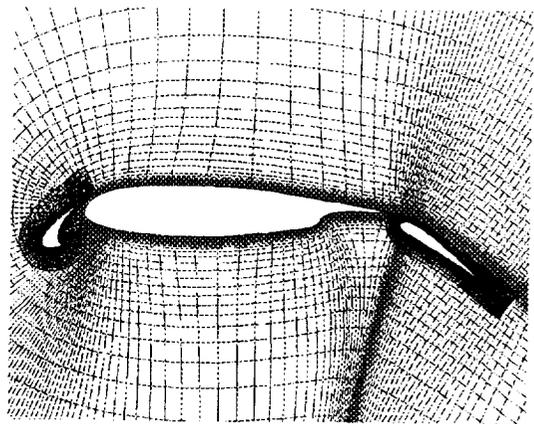


Figure 2 Grid around three-element airfoil (every other point shown for clarity).

### Governing Equations and Numerical Methods

In order to obtain solutions for the computational training data, 54 configurations are solved at 10 different angles of attack. This study is performed for two-dimensional flows since it is less computationally intensive than three-dimensional problems and allows the investigation of many parameters. The incompressible Navier-Stokes equations in two-dimensional generalized coordinates are solved using INS2D-UP (Rogers and Kwak, 1990, 1991) flow solver. This code has been used extensively to predict high-lift multi-element airfoil flows. INS2D-UP uses an artificial compressibility approach to couple the mass and momentum equations. The convective terms are differenced using a third-order accurate upwind biased flux-splitting. The equations are solved using a generalized minimum residual implicit scheme. Since the flow is turbulent, the Spalart-Allmaras (Spalart and Allmaras, 1992) turbulence model is used in this study for closure. The Spalart-Allmaras turbulence model has been successfully used to compute flowfields associated with high-lift multi-element airfoils (Rogers, 1993 and Dominik, 1994).

### Maximum Lift Criteria

The determination of maximum lift is one of the most important results of any high-lift wing design study. Figure 3 shows the computed lift coefficient versus angle of attack for one high-lift setting. The solid symbols show that the computed solutions do not display the characteristic increase in  $C_l$  with increasing angle of attack up to  $C_{l_{max}}$ . For angles of attack beyond that point, the lift coefficient should decrease. Valarezo and Chin (1994) reported a hybrid method that couples cost-effective computational fluid dynamics technology with empirically-observed phenomenon in order to predict maximum lift ( $C_{l_{max}}$ ) for complex multi-element wing geometries. Their semi-empirical  $C_{l_{max}}$  criteria for multi-element airfoils or wings, designated the pressure difference rule, is applied to the computational training data set. The pressure difference rule

$\delta_f = 25.0$ deg.		$\delta_f = 29.0$ deg.		$\delta_f = 39.5$ deg.							
gap	0.4	1.0	1.5	gap	0.4	1.0	1.5	gap	0.4	1.0	1.5
1.5	1	10	19	1.5	2	11	20	1.5	3	12	21
2.1	4	13	22	2.1	5	14	23	2.1	6	15	24
2.7	7	16	25	2.7	8	17	26	2.7	9	18	27

a) Method 1

gap	0.4	1.0	1.5	gap	0.4	1.0	1.5	gap	0.4	1.0	1.5
1.5	1	10	19	1.5	(2)	11	(20)	1.5	3	12	21
2.1	(4)	13	(22)	2.1	5	14	23	2.1	(6)	15	(24)
2.7	7	16	25	2.7	(8)	17	(26)	2.7	9	18	27

b) Method 5

gap	0.4	1.0	1.5	gap	0.4	1.0	1.5	gap	0.4	1.0	1.5
1.5	1	10	19	1.5	(2)	11	(20)	1.5	3	12	21
2.1	4	13	(22)	2.1	5	14	(23)	2.1	6	15	24
2.7	7	16	25	2.7	8	17	26	2.7	(9)	18	27

c) Method 9

**Figure 7** Computational cases that are used to train the neural networks.

data is presented to the neural networks to learn) required to train the neural networks. The results of this learning curve are presented by Greenman (1998). It was determined that 250 iterations was optimal for this study.

Even though the computational database that is used is sparse, a study (Greenman, 1998) was conducted to see how much further the training set can be reduced and still allow the neural networks to predict within the acceptable error. Several subsets of the computational data were used to train the neural nets. It was shown that carefully selecting configurations to omit from the training set, neural networks can be trained with only 50-74% of the entire data set to accurately predict the aerodynamic characteristics of a multi-element airfoil (Greenman, 1998). Method 1 designates the training set which contains all the computed data (Figure 7). Figure 7 shows additional training methods that are successful in training the neural networks and that are presented in this paper. Here, the shaded boxes represent the cases that are in the training set whereas the numbers in the white boxes and in the parentheses are the cases that are omitted from the training set.

### High-Lift Flap Setting Optimization

The high-lift system is optimized by maximizing the lift

**Table 1** Design Space for  $\delta_s = 6.0$  degrees

Design Variables	Lower Bound	Upper Bound
$\delta_f$	25.0	38.5
$gap_f$	1.50	2.70
$overlap_f$	0.40	1.50
$\alpha$	0.0	10.0

coefficient. The design variables in this study are chosen to be the flap deflection, gap, overlap, and angle of attack. The bounds on the design space (shown in Table 1) are chosen to be the same as the design space that are used to train the neural networks with the exception that for optimization cases without constraints, the angle of attack is bounded to  $\alpha = 10.0^\circ$  since this is near the range where maximum lift is predicted to occur by the pressure difference rule for most of the configurations. To start the optimization, the initial values of the design variables are arbitrarily chosen.

**Method 9.** Method 9 is used to train the neural networks which are integrated with the optimizer. Method 9 contains only 74% of the entire configurations in the training set (Figure 7). Five different optimization runs are shown in Table 2. Each of these runs has different initial or starting values (orig) of the design variables (DV). Gradient based optimizers do not guarantee that the maximum which is found is the global maximum of the design space; it only guarantees an improvement. Thus, different starting values of the design variables are used to search the entire design space. The first optimization run, 9-A, has the initial design variables set to the lower bounds. Whereas, the second run, 9-B, has the initial values set to the upper bounds of the design space. In the third run, 9-C, the initial conditions are set to the average value of the lower and upper bounds. The last two runs have arbitrary initial values to test different regions of the design space. With this optimization procedure, the design space can be easily searched with several optimization runs because each run only requires several seconds of CPU time. A total of 28.6 CPU seconds are used for these five optimization runs.

In this study, the optimizer found 2 different maximums. The smaller of the two maximums is found using the initial design variables of Runs 9-B through 9-E. The modified high-lift rigging is  $\delta_f = 38.5^\circ$ ,  $gap_f = 2.04\%$ ,  $ol_f = 1.50\%$ , and  $\alpha = 10.0^\circ$  and has  $C_l = 4.11$ . The other maximum for this particular study is just slightly higher at  $C_l = 4.13$ . The modified values of the design variables for this case are  $\delta_f = 38.5^\circ$ ,  $gap_f = 2.01\%$ ,  $ol_f = 0.56\%$ , and  $\alpha = 10.0^\circ$ . The flap deflec-

**Table 2 Optimization Results with Method 9 as the Training Set**

Run	DV	orig	mod	$C_l$ orig NN	$C_l$ orig INS2D	$\Delta\%$ orig	$C_l$ mod NN	$C_l$ mod INS2D	$\Delta\%$ mod	$\Delta C_p$ diff mod INS2D	CPU (sec)
9-A	$\delta_f$	25.0	38.5	2.04	2.04	0.0	4.13	4.03	2.48	-14.8	6.9
	gap <sub>f</sub>	1.50	2.01								
	ol <sub>f</sub>	0.40	0.56								
	$\alpha$	0.0	10.0								
9-B	$\delta_f$	38.5	38.5	3.54	3.56	-0.56	4.11	4.00	2.75	-14.5	3.3
	gap <sub>f</sub>	2.70	2.04								
	ol <sub>f</sub>	1.50	1.50								
	$\alpha$	10.0	10.0								
9-C	$\delta_f$	32.0	38.5	3.19	3.20	-0.31	4.11	4.00	2.75	-14.5	6.9
	gap <sub>f</sub>	2.10	2.04								
	ol <sub>f</sub>	0.95	1.50								
	$\alpha$	5.0	10.0								
9-D	$\delta_f$	30.0	38.5	3.02	2.96	2.02	4.11	4.00	2.75	-14.5	5.5
	gap <sub>f</sub>	1.90	2.04								
	ol <sub>f</sub>	0.75	1.50								
	$\alpha$	4.0	10.0								
9-E	$\delta_f$	27.0	38.5	2.51	2.47	1.62	4.11	4.00	2.75	-14.5	6.0
	gap <sub>f</sub>	2.10	2.04								
	ol <sub>f</sub>	0.50	1.50								
	$\alpha$	2.0	10.0								

tion for both instances is optimal at the upper bound. The modified gaps are free variables (the variable lies between the upper and lower bounds) and close to each other, whereas the overlaps are quite different. The smaller maximum has the overlap at the upper bound whereas the larger maximum has it as a free variable. Both configurations have the angle of attack to be optimal at the upper bound.

The accuracy of the neural network prediction is tested for both the initial and modified configurations by generating the appropriate grid and computing the INS2D solution. Then the

predicted and computed  $C_l$  are compared and the percent difference ( $\Delta\%$ ) is shown in Table 2. The initial configurations have lower errors than the modified configurations. In Run 9-A there is zero error and only one case has an error greater than 2%. Modified configurations have prediction errors greater than 2%. The pressure difference rule is applied to these cases to determine if the modified configurations have a  $C_{p_{diff}}$  less than the acceptable value. Examining the outcome, shows that the pressure difference exceeds the allowable value of  $C_{p_{diff}} = -13.0$ . All the pressure differences are equal to or greater than  $C_{p_{diff}} = -14.5$ . Some CFD training data may be

**Table 3 Constrained Optimization Results for Method 9 as the Training Set**

Run	DV	orig	mod	$C_l$ orig NN	$C_l$ orig INS2D	$\Delta\%$ orig	$C_l$ mod NN	$C_l$ mod INS2D	$\Delta\%$ mod	$\Delta C_p$ diff mod NN	$\Delta C_p$ diff mod INS2D	CPU (sec)
9-C- $\Delta C_p$	$\delta_f$	32.0	37.5	3.19	3.20	-0.31	3.94	3.86	2.07	-13.0	-13.0	27.3
	gap <sub>f</sub>	2.10	2.08									
	ol <sub>f</sub>	0.95	0.40									
	$\alpha$	5.0	9.0									
9-C- opt	$\delta_f$	32.0	38.5	3.18	3.20	-0.63	3.94	3.92	0.51	-13.0	-13.4	26.1
	gap <sub>f</sub>	2.10	1.5									
	ol <sub>f</sub>	0.95	0.4									
	$\alpha$	5.0	8.30									

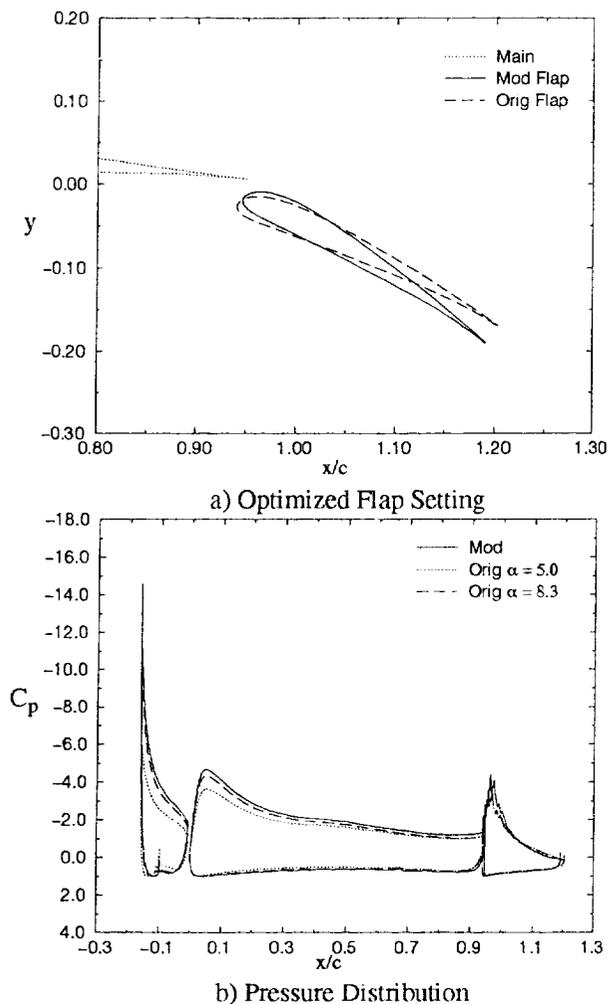
non-physical at the upper bound of the angle of attack since the bound on angle of attack is chosen to be an average value of where maximum lift occurs. Consequently, the neural networks are not properly trained to predict the aerodynamics in this range.

**Constrained Optimization.** In order to test whether the accuracy would get better if the modified configurations were restricted within the empirically predicted pre-stall range, the upper bound on the angle of attack design variable is removed. Instead a constraint is placed on the value of the pressure difference,  $C_{p_{diff}} \geq -13.0$ . An additional neural network is trained with flap deflection, gap, overlap, and angle of attack to predict the pressure difference. In this case, the entire training data is used to compute  $C_{p_{diff}}$ , whereas the neural networks that compute the aerodynamic coefficients are trained with data only including pre-stall data that is predicted by the pressure difference rule. The design variables of the optimization runs remain the same as does the objective function. The results of the case that found the best improvement by the optimizer is shown in Table 3 for Run 9-C- $\Delta C_p$ . The modified design variables are  $\delta_f = 37.5^\circ$ ,  $gap_f = 2.08\%c$ ,  $ol_f = 0.40\%c$ , and  $\alpha = 9.0^\circ$ . The modified angle of attack is lower than in the previous case that specified the upper bound to be  $\alpha = 10.0^\circ$ . The neural network predicted the pressure difference value to be exactly what is calculated with the INS2D solution, and predicted the modified lift coefficient to be higher than 2% the actual INS2D value.

To further reduce the prediction error in the modified lift coefficient, the INS2D data from this optimal case is added to the training data. The neural networks are then re-trained with this additional information in hope that it will improve the accu-

racy. Again, the neural network that predicts the lift coefficient is trained with the data set that includes the data points that are at or below the maximum lift. The optimization runs are again constrained and the best improvement is shown in Table 3 denoted by Run 9-C-opt. The values of the modified design variables are different for the flap deflection, gap, and angle of attack and are the same for the overlap that in the previous case. The modified lift coefficient predicted by the neural network happens to be the same as in the previous optimization run, however, the INS2D value of the modified coefficient is different and the error is reduced to only 0.51%. Thus, by constraining the design space that the optimizer is allowed to search and by adding one data point near maximum lift to the training data, the prediction error is reduced and all constraints are met. The predicted and actual pressure difference are close and differ by only 0.4. It should be noted that the CPU time required to run a constrained optimization run is increased, however, it is still less than 30 seconds as shown in Table 3.

To get a better understanding of the flow physics, the pressure distribution of the modified and original configurations for optimization Run 9-C-opt are examined. Figure 8a shows the modified and original flap positions in relation to the main element trailing edge. Figure 8b shows the pressure distribution of the slat, main, and flap elements in a solid line for the modified configuration. The original configuration was initially at  $\alpha = 5.0^\circ$  (plotted in a dotted line) but in order to compare the pressure distributions, the original configuration is also plotted at  $\alpha = 8.3^\circ$  (in a dashed line). The basic shape of the  $C_p$  curves are similar for all elements for both configurations. The flow is attached for all elements. The suction pressure on the modified elements are clearly larger than the original configuration result-

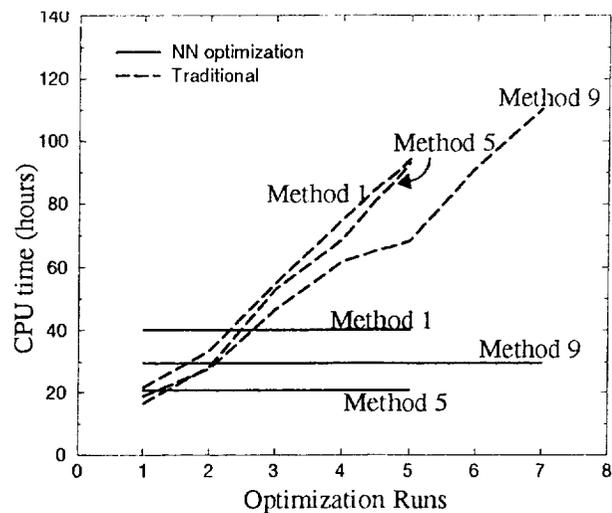


**Figure 8** Optimization results for run 9-C-opt (flap settings denoted in Table 3).

ing in greater lift. There are interesting features on the original and modified flap elements. The sharp spike at the trailing edge occurs from the sharp point at the trailing edge of the flap geometry. The numerical grid comes to a sharp corner at the trailing edge, the flow must accelerate at this point causing the pressure to drop. The multiple spikes that are located at the leading-edge of the flap element are associated with the original definition of the geometry. The flap at this region is faceted due to the high curvature. The pressure spikes are representative of what the flow is actually doing. The flow is turning around at these facets and accelerating.

### BENEFITS OF NEW PROCESS

The aerodynamic design space of a multi-element airfoil is very complex and may have many local maximums and minimums. When a gradient-based optimizer is used to search the design space, many starting points need to be examined in order



**Figure 9** Comparison of CPU time required for traditional and neural network optimization procedures.

to find the greatest improvement. This can be very computationally expensive in traditional optimization.

### Computational Resources

The advantage of using neural networks in the optimization process versus the traditional optimization process is the turn around time and the CPU time that is saved for many optimization runs. In the traditional optimization process, every time that the design variables are perturbed, the gradient needs to be calculated to determine the search direction. In order to calculate the gradient, a grid needs to be generated and the aerodynamic coefficients must be calculated by solving the flowfield with INS2D. Even though, the traditional optimization method will have shorter turn around time and CPU time when doing one or two optimization runs, there is no guarantee that one or two optimization runs will find the global maximum. On the contrary, the neural networks will have less overall turn around time and CPU time for many optimization runs and there is no major increase in overall turn around or CPU time for additional runs. Once the neural networks are trained, only 5-10 seconds are required for each additional optimization run. The CPU time that is used in this optimization study for the different training methods used is shown in Figure 9. Also plotted in this figure are the calculated CPU time that would have been used in the traditional optimization process. The CPU time for the traditional method is estimated by using the same number of function calls that is used in the neural network optimization procedure. Then for each iteration it is estimated that the CPU time will consist of 4.3 seconds to generate a grid and 600 seconds (on a Cray C90) for each flow solution. If more than three optimization runs are executed, then the neural network optimization procedure should be used. The neural network optimiza-

**Table 4 Neural Network Optimization Procedure Cost**

Method	Generating Training Set CPU (hours)	Training CPU time (seconds)	Optimization CPU time (seconds)	Total Cost (dollars)
1	40.08	281.0	263.0	6466.50
5	20.78	207.0	93.5	3353.50
9	29.69	232.0	176.6	4790.16

tion procedure curves are nearly flat. Thus, the major contributor to the CPU time in the neural network optimization is training the neural networks to learn and to predict the aerodynamics of the airfoil. Many more optimization runs can be executed with this procedure without requiring large additional amount of CPU time. On the other hand, the traditional optimization procedure will continue to increase at a fairly linear rate as shown.

### Cost Analysis

Another advantage of using the neural network optimization procedure is reduction of cost. There are many factors contributing to the total cost of a research job including the cost of the engineer support, computer resources, and wall clock turn around time. One of the largest contributors to turn around time is waiting for a computer job to be completed especially if the job executes within a batch queue. The average turn around time for the computers used in this study at the Numerical Aerospace Simulation Facility (NAS) at NASA Ames Research Center is 23.45 hours for an eight-hour queue job.

To calculate the cost that is related to the two types of optimization procedures considered, it is assumed that an experienced engineer is executing both optimization processes. This engineer is familiar with the different components to each process such as grid generation, flow simulation, neural networks, and optimization. The set-up time is assumed to be equal for both processes. The engineer is a full time equivalent of \$200,000 per year and there are 2080 working hours in a year. Thus, there is a charge of \$96.15 per hour for an engineer. Another expense which must be considered is computer resources. For this comparison, assume the cost of a computing hour is \$39.00.

First, the cost of the neural network optimization procedure is calculated. A grid is generated for each configuration included in the training method and solutions are calculated for 10 different angles of attack for each configuration. The grid generation requires 4.3 CPU seconds per grid and 269.0 CPU seconds per flow solution (the convergence time is low since these solutions are below maximum lift). The CPU time

**Table 5 Traditional Optimization Procedure Cost**

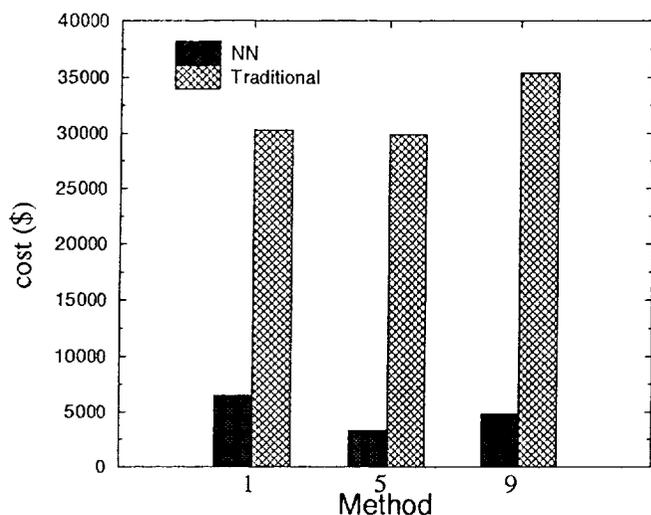
Method	CPU Hours	Number of 8 hour jobs	Wall Clock (hours)	Total Cost (dollars)
1	94.31	11.79	276.45	30,261.21
5	93.13	7.83	272.96	29,876.98
9	110.31	13.79	328.30	35,391.83

required to train each method and used to optimize all five optimization runs (see Table 4) must also be added to the total wall clock time and the charged CPU time. The total cost of the neural network optimization procedure is shown Table 4. The major element in the cost is the time and computer resources required to set-up the training matrix data. Consequently, it is very important to determine the level of prediction accuracy that is required and to choose the proper method to train the neural networks.

Second, the cost of the traditional optimization procedure is calculated with the same assumptions. The wall clock time and the CPU hours charged are calculated based on the number of iterations (or gradient calls) that are made by the optimizer for each optimization run. For each method that is used in the neural network optimization procedure, the traditional optimization cost is calculated for the same five optimization starting runs. The total turn around (wall clock) time that the engineer waits for the job to be finished is multiplied by \$96.15 and is added to the total CPU hours that are charged. The traditional optimization procedure is performed on the Cray computer in the batch queue. This is one of the reasons that the cost is higher than the neural network optimization procedure as shown in Table 5.

The total costs are compared in Figure 10 for the two optimization procedures. For five optimization runs for each training method, the neural network optimization procedure does cost less. Again, if only one or two optimization runs are performed, then the traditional optimization procedure would cost less, however, for multiple runs, the neural network optimization procedure uses less resources. The biggest advantage now is that many more optimization runs can be performed with the neural network optimization procedure while only adding seconds to the CPU time and turn around time.

The neural network optimization procedure should be used for design because several designs with different constraints or design space can be considered without driving the cost and turn around time up. Also, once a design is chosen, the design space can be altered and the optimization procedure can now be performed again with minimum additional cost and turnaround time.



**Figure 10** Comparison of total cost for the neural network and traditional optimization procedure.

## CONCLUSIONS

An enhanced design process was developed which integrates neural network and optimizer technologies together with a computational database. The process is modular, allowing insertion of emerging neural network, optimization, and CFD technologies within its framework. This design process was tested for a typical high-lift design problem to optimize flap rigging for maximum lift. Initial studies showed that although optimization could be conducted using a sparse training dataset, unconstrained optimization of the high-lift system produced unacceptably high errors. Due to the complexity of the high-lift flow physics near the maximum lift condition, an empirically based constraint, which identifies configurations at the maximum lift condition within the computational database, was required in order to achieve accurate neural net predictions for this design problem. Using the empirical constraint together with an iterative optimization procedure which re-inserted the optimized configuration into the training database and repeated the optimization produced an optimal configuration with only 0.5% error. A cost analysis was conducted by comparing the optimization with neural networks procedure to the traditional optimization procedure. It was found that the optimization with neural networks procedure resulted in a reduction of turnaround time, CPU time, and cost if more than two optimization runs were conducted. Using the optimization procedure, the average cost reduction is 83%.

## ACKNOWLEDGMENTS

This work would not have been possible without the helpful discussions and suggestions from Dr. James C. Ross, Dr. Stuart E. Rogers, and Dr. Charles C. Jorgensen at NASA Ames Research Center.

## REFERENCES

- Chan, W. M., Chui, I. T., and Buning, P. G., 1993, "User's Manual for the HYPGEN Hyperbolic Grid Generator and the HGUI Graphical User Interface," NASA TM 108791.
- Dominik, C., 1994, "Application of the Incompressible Navier-Stokes Equations to High-Lift Flows", AIAA Paper 94-1872.
- Faller, W. E. and Schreck, S. J., 1995, "Real-Time Prediction of Unsteady Aerodynamics: Application for Aircraft Control and Maneuverability Enhancement," *IEEE Transactions on Neural Networks*, Vol. 6, No. 6, pp. 1461 - 1468.
- Gill, P. E., Murray, W., Saunders, M. A., and Wright, M. H., 1994, "User's Guide for NPSOL 5.0: A Fortran Package for Nonlinear Programming," Dept. of Operations Research, Stanford University, TR SOL 94, Stanford, CA.
- Greenman, R. M., 1998, "Two-Dimensional High-Lift Aerodynamic Optimization Using Neural Networks", *Stanford University Ph. D. Dissertation*.
- Greenman, R. M., Cheung, S., and Tu, E. L., 1998, "Coupled Navier-Stokes and Optimizer Analysis of a Transonic Wing," *Journal of Aircraft*, Vol. 35, No. 3, pp. 362-369.
- Jorgensen, C. C. and Ross, J. C., "System and method for Modeling the Flow Performance Features of an Object," U. S. Patent No. 5,649,064, July 1997.
- LaMarsh, W. J., Walsh, J. L., and Rogers, J. L., 1992, "Aerodynamic Performance Optimization of a Rotor Blade Using a Neural Network as the Analysis," AIAA Paper 92-4837.
- McMillen, R. L., Steck, J. E., and Rokhsaz, K., 1995, "Application of an Artificial Neural Network as a Flight Test Data Estimator," *Journal of Aircraft*, Vol. 32, No. 5, pp. 1088 - 1094.
- Norgaard, M., Jorgensen, C. G., and Ross, J. C., 1997, "Neural-Network Prediction of New Aircraft Design Coefficients," NASA TM 112197.
- Rai, M. M. and Madavan, N. K., 1998, "Application of Artificial Neural Networks to the Design of Turbomachinery Airfoils," AIAA Paper 98-1003.
- Rogers, S. E., 1993, "Progress in High-Lift Aerodynamic Calculations," AIAA Paper 93-0194.
- Rogers, S. E., 1997, "Manual for the OVERMAGG Script System," NASA Ames Research Center.
- Rogers, S. E. and Kwak, D., 1990, "An Upwind Differencing Scheme for the Time Accurate Incompressible Navier-Stokes Equations," *AIAA Journal*, Vol. 28, No. 2, pp. 253 - 262.
- Rogers, S. E. and Kwak, D., 1991, "An Upwind Differencing Scheme for the Steady State Incompressible Navier-Stokes Equations," *Journal of Applied Numerical Mathematics*, Vol. 8, pp. 43 - 64.
- Ross, J. C., Jorgenson, C. C., and Norgaard, M., 1997, "Reducing Wind Tunnel Data Requirements Using Neural Networks," NASA TM 112193.
- Spalart, P. R. and Allmaras, S. R., 1992, "A One-Equation Turbulence Model for Aerodynamic Flows," AIAA Paper 92-

0439.

Steck, J. E. and Rokhsaz, K., 1997, "Some Applications of Artificial Neural Networks in Modeling of Nonlinear Aerodynamics and Flight Dynamics," AIAA Paper 97-0338.

Storms, B., 1997, Private communication.

Suhs, N. E. and Tramel R. W., 1991, "PEGSUS 4.0 User's Manual," AEDC-TR-91-8.

Valarezo, W. O. and Chin, V. D., 1994, "Method of Prediction of Wing Maximum Lift," *Journal of Aircraft*, Vol. 31, No. 1, pp. 103-109.

Ying, S. X., 1996, "High Lift Challenges and Directions for CFD," *AIAA/NPU AFM Conference Proceedings*, China.